

<論 文>

3차원 자유곡면 가공에 있어서의 공구간섭방지에 관한 연구

양균의* · 박윤섭** · 이희관**

(1994년 10월 10일 접수)

A Study on the Avoidance of Tool Interference in Free Form Surface Machining

Gyun-Eui Yang, Yeun-Sep Park and Hi-Koan Lee

Key Words : SSI Algorithm(곡면/곡면 교차곡선 알고리즘), Tool Interference(공구간섭), Free Form Surface(자유곡면), Offset Surface(오프셋곡면), Cutter Location Data(공구경로)

Abstract

Tool interference is one of the most critical problems in sculptured surface machining. When machining cavities and concaves, the tool frequently overcuts the portions of the surface, which cause inaccuracy in machining. So tool interference-free paths must be generated for rough cutting more efficiently. In this paper a software using SSI(Surface/Surface Intersection) algorithm is developed for eliminating tool interference which occurs in an offset surface in 3-dimensional free form surface modeling. This work consists of two stages: using the offset data, the intersection curves are rapidly checked by this algorithm at the first stage. CL(cutter location) data are obtained by deleting the loop section of intersected offset patches at the second stage. This algorithm can reduce the amount of memory required to store machining data and also easily check region which have the possibility of intersection. Also, This software is verified to be useful in machining a curved object on a DNC milling machine

1. 서 론

가전제품, 항공기, 자동차 등에 사용되는 금형을 제작하기 위해서는 효율적인 자유곡면의 modeling을 해야 하며, 이 모델링이 수행된 후에는 공구경로 계산, post-processing, 황삭계획, 정삭계획이 수행되어야만 한다. 특히 위와 같은 공정계획에 의해 자유곡면을 가공할 때에 가장 심각하게 발생하는 문제가 공구간섭현상이다. 이와 같은 간섭현상이 발생하면 공구가 원치 않는 부분을 가공하게 되

어 금형 가공물의 정도를 크게 저하시키게 된다. 곡면의 NC가공에 있어서 공구간섭은 대부분 가공곡면의 오목한 지역에서 곡률반경이 공구의 반경보다 작은 경우에 발생하게 된다. 간섭현상을 피하기 위해서는 곡면상의 최소 곡률반경보다 작은 공구를 사용하는 것을 생각해 볼 수 있으나 가공시간과 가공효율 향상의 측면에서 보다 큰 반경의 공구에 의한 황삭가공이 필요하므로 공구반경에 알맞은 공구경로의 개발은 필수적이라 하겠다. 한편 기존 CAD/CAM 시스템에서 공구간섭 검색 및 제거 routine이 있어도 상업적인 가치때문에 공개되지 않는 것이 현실이다. 기존 간섭에 관한 논문이 나와 있지만 보다 신속하게 가공하기 전에 간섭을 검

*정희원, 전북대학교 기계공학과

**전북대학교 대학원 기계공학과

색하고 제거할 수 있는 알고리즘의 개발이 자유곡면 금형가공에 있어서 정도향상 및 생산성향상 측면에서 필수적이라고 볼 수 있다.

자유곡면의 NC가공에 있어서 기존 공구간섭에 대한 논문을 살펴보면, 공간상에서 공구의 위치를 규정하기 위하여 가공곡면의 part surface, 공구 진행방향 지정의 drive surface, 공구정지위치 지정의 check surface를 지정하여 각각의 곡면에 대하여 공구가 놓이는 위치를 지정하는 APT방법,⁽¹⁾ Cartesian 또는 parametric 공간상에서 공구경로를 계획하고 이 경로에 해당하는 곡면상의 곡선을 구한 다음 이를 여러 개의 선분들로 적절히 보간하여 CC(cutter contact) data를 얻어서 이 CC data에서 간섭영역에 속하는 부분들을 제거함으로써 간섭을 방지하는 CL(cutter location) data를 얻는 방법,⁽²⁾ Duncan과 Mair^(3,4)에 의해 제안된 가공곡면을 삼각형의 면으로 형성된 다면체로 근사하고 공구는 각 삼각형 면의 도심에 접하게 하여 공구가 주변의 삼각형들과 간섭되는지를 검사하고 간섭의 경우 공구를 Z축(공구축) 방향으로 들어올리도록 하는 방법, CC data로 다면체 근사모형을 형성하여 주고 간섭을 검색하는 방법⁽⁵⁾ 그리고 공구의 inverse offset을 이용해 공구간섭이 제거된 공구경로 생성법⁽⁶⁾ 등이 연구되었다. 이들 방법들은 NC정보들이 가공전에 미리 준비되어야 하므로 대부분 계산시간이 많이 들고 가공 data의 저장에 큰 기억용량을 필요로 하는 단점이 있다.

본 연구에서 이러한 단점을 개선하기 위하여 제안하고자 하는 방식은 간섭검사에 필요한 시간과 기억용량의 소모를 줄이고자 교차곡선의 시작점을 찾는 격자법⁽⁷⁾과 교차곡선 상의 점을 찾는 추적법⁽⁸⁾을 접목시켜 SSI(surface/surface intersection) 알고리즘을 구현하여 offset 곡면간의 교차를 신속하게 검색할 수 있도록 Bezier 특성다각형 성질을 이용하여 교차가능 영역을 판정하고 계산속도를 높이고자 하였다. 여기서 얻어지는 교차곡선사이의 간섭영역 제거로 간섭없는 공구경로를 산출해 냄으로써 가공 data의 저장에 필요한 기억용량의 크기도 줄일 수 있었으며 공구경로 선정에도 유연성을 가지게 하였다. 또한 본 연구에서 개발된 공구간섭검색 및 제거 알고리즘을 검증하기 위해 본 실험실에서 보유한 3축 DNC milling machine에 의하여 실제 가공물을 택해 간섭없는 자유곡면 가공을 수행하였다.

2. Bezier곡면의 특성 및 입력 Data Point를 통과하는 Bezier 곡면의 형성

본 논문에서 자유곡면 표현기법은 Bezier 곡면을 이용하였다. 이는 뒤에서 설명하였지만 Bezier의 특성다각형 성질을 이용하여 교차가능 영역을 쉽게 판정할 수 있기 때문이다. Bezier 곡면의 근사는 주어진 조정점으로 이루어지는 Bezier 특성다각형 안에서 이루어지게 되는데 주어진 조정점을 구하기 위해서는 각 patch를 구성하는 4개의 모서리 점에서 Bezier 곡면과 Ferguson 곡면을 일치하게 되므로 Ferguson 곡면생성 표현식을 이용하여 주어진 각 점에서의 u, v parameter방향으로 접선벡터와 꼬임벡터를 구해 4개의 모서리점에서 Ferguson 곡면방정식과 Bezier 곡면방정식을 일치시켜 얻어진 Bezier 조정점으로 인해 Bezier 곡면을 생성시킨다.

3차 Bezier 곡면과 Ferguson 곡면을 수식으로 표현하면 각각 식(1)과 (2)의 간단한 행렬식으로 표현된다.⁽¹⁾

$$r(u, v) = [1, u, u^2, u^3] MBM^T [1, v, v^2, v^3]^T \quad (1)$$

$$r(u, v) = [1, u, u^2, u^3] CQC^T [1, v, v^2, v^3]^T \quad (2)$$

$r(u, v)$ 는 2개의 변화하는 매개변수로 표현되는 곡면점의 공간좌표이다. 단 u, v 는 실수 매개변수이고 $0 \leq u, v \leq 1$ 이다.

M 는 Bezier patch를 특정짓는 상수행렬이고 B 는 Bezier patch의 조정점행렬이다. C 는 Ferguson patch를 특정짓는 상수행렬이고 Q 는 Ferguson patch의 경계곡선 특성치행렬이다. 여기서 각 patch를 구성하는 4개의 모서리점에서 Bezier 곡면과 Ferguson 곡면은 일치하게 되므로 그 결과 얻어지는 Bezier 조정점은 다음과 같이 표현된다.

$$B = (M^{-1}C)Q(M^{-1}C)^T \quad (3)$$

본 논문에서는 Bezier 곡면의 불록페포성을 교차가능성이 있는 patch를 지정하는데 이용한다.

3. Offset 곡면과 공구간섭

3.1 Offset 곡면

Ball mill을 이용하여 가공을 하는 경우와 같이

곡면상의 한 점에 구를 접하도록 위치시키면 접점 r_c 에서 구의 중심을 향하는 선분을 정의할 수 있는데, 이 선분의 길이를 1로 잡으면 점 r_c 에서의 곡면의 단위법선벡터를 정의할 수 있다.

$$N = \pm \left(\frac{\partial r}{\partial u} \times \frac{\partial r}{\partial v} \right) / \left| \frac{\partial r}{\partial u} \times \frac{\partial r}{\partial v} \right| \quad (4)$$

NC가공에서 공구의 위치를 나타내는 좌표값을 CL data라고 부르는데 ball mill에서는 공구의 바닥 끝점을 기준으로 삼는다. Ball mill의 반경을 R 라 하고 공구의 회전축방향(공구 끝에서 기계스핀들을 향하는 방향)을 단위벡터 u 로 표시하면, CL-data r_L 은 곡면상의 접촉점 r_c 로부터 다음과 같이 구해진다.

$$r_L = r_c + R(N - u) \quad (5)$$

3축 NC가공에서는 $u = (0, 0, 1)$ 이다.

구해진 이 CL data들은 공구의 경로를 구성하며 이 경로를 따라 가공이 수행될 수 있다. 이때 공구의 중심점인 offset점들의 거동을 살펴보면 공구의 바닥 중심점들인 CL data와 똑같은 경로를 보이는 것을 알 수가 있다. 이는 3축 가공의 경우 공구 offset data와 CL data가 Z축에 양의 수직방향으로 공구반경만큼 이동한 관계를 가지기 때문이다.

Fig. 1에 CL data와 offset data를 보였다. 가공 곡면상의 한 점 r_c 에 공구가 접해 있을때 그 접점을 CC점(cutter contact point)이라 하고, 이때의 공구중심점을 'offset점'이라 하며 이 offset점이 이루는 궤적이 offset 곡면을 형성하며 이 offset점을 수식으로 표시하면 다음과 같다.

$$P_{off}(u, v) = P(u, v) + \frac{\frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v}}{\left| \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v} \right|} R \quad (6)$$

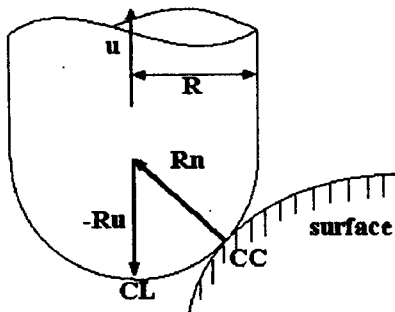


Fig. 1 CL data and offset data

여기서, $P(u, v)$ 는 직교좌표벡터 (x, y, z) 이다.

$P_{off}(u, v)$ 의 1차 미분항은 다음과 같이 표시된다.

$$\frac{\partial P_{off}}{\partial u} = \frac{\partial P}{\partial u} + \left\{ \frac{\frac{\partial N}{\partial u}}{|N|} - \frac{N \left(N \frac{\partial N}{\partial u} \right)}{|N|^3} \right\} R$$

$$\frac{\partial P_{off}}{\partial v} = \frac{\partial P}{\partial v} + \left\{ \frac{\frac{\partial N}{\partial v}}{|N|} - \frac{N \left(N \frac{\partial N}{\partial v} \right)}{|N|^3} \right\} R$$

여기서,

$$N = \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v}$$

$$\frac{\partial N}{\partial u} = \frac{\partial^2 P}{\partial u^2} \times \frac{\partial P}{\partial v} + \frac{\partial P}{\partial u} \times \frac{\partial^2 P}{\partial u \partial v}$$

$$\frac{\partial N}{\partial v} = \frac{\partial^2 P}{\partial u \partial v} \times \frac{\partial P}{\partial v} + \frac{\partial P}{\partial u} \times \frac{\partial^2 P}{\partial v^2}$$

여기서, 원래의 곡면 $P(u, v)$ 는 u, v 방향으로 각각 2차 미분가능하여야 함을 알 수 있다.

3.2 공구간섭

이와 같이 offset 곡면을 형성하여 주면 원래의 곡면에서는 보이지 않던 교선 loop가 발생하는 경우가 생기게 되고 그 loop를 제거해 주지 않으면 가공시 과절삭이 발생하게 된다. Fig. 2는 곡률반경보다 큰 반경을 가진 공구로 가공을 수행할 경우의 간섭 발생을 보여준다.

이와 같이 loop가 발생하여 과절삭이 발생하는 공구경로가 생기지 않게 하는 것이 간섭방지인데, 이를 위해서는 애초에 loop가 생기지 않는 공구를 선택하면 되겠다. 이를 위해서는 전 가공면에 있어서 곡률을 계산해 가며 곡면을 만들어 주어 얻어진 최소 곡률반경보다 작은 공구반경을 가진 공구를 선택해 주면 되겠으나, 황삭가공의 경우와 같이 효

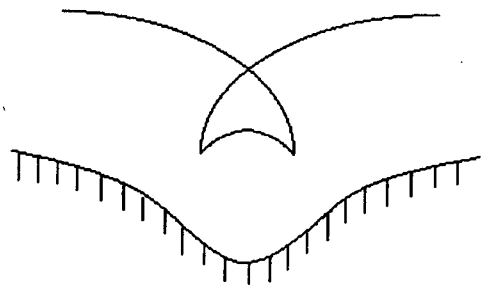


Fig. 2 Offset path with interference

율을 위해 큰 공구반경을 가진 공구의 선택이 불가피한 경우에는 offset면에서의 loop발생이 불가피하게 된다. 기존의 경우 이 간섭 loop의 발견은 모니터상에 나타난 그래픽으로 확인하거나 또는 각 가공면 위의 각 점에서의 곡률계산이 필요하다. 그러나 모니터 상에 확실히 나타나지 않을 정도의 작은 꼬임인 경우 곡률계산의 방법이 더 믿을 만하겠

다. 그러나 이 곡률 계산은 많은 계산 양이 필요하므로 본 논문에서는 공구가 정해지면 그 공구로 인한 간섭여부를 신속하게 알 수 있는 간단한 방법을 사용한다. 이때 간섭을 일으키는 부분은 loop의 교점사이의 data들이므로 이들을 제거해 주면 공구간섭이 없는 CL data를 얻는 것이 가능하다. 그런데 이 loop의 교점은 Cartesian 경로를 따라 가공하는 경우를 제외하고는 반드시 2차원적으로 교차한다고 보장할 수가 없다. 따라서 parametric 경로를 따라 가공을 수행하는 경우에는 실제 교점을 찾기가 복잡한 3차원 loop가 발생하기가 쉽다. 이 경우 SSI 알고리즘을 이용해 교차하는 곡면간의 교선을 구하면, 이때 구해지는 교선은 교차하는 면들 전체에 걸쳐 계산되고 각각의 patch구간에 있어서의 구간 parameter값으로 정의되므로 3차원 loop가 생성되어도 그 교점의 parametric 경로상에서의 값이 정확하게 구해질 수 있다. 또 하나의 공구반경에 대한 교선을 구해 놓으면 경로간격이 달라지더라도 다시 계산을 수행할 필요가 없이 이미 구한

교선상에서의 새 공구경로 간격값을 넣어주는 것으로 교점 data를 구해낼 수 있는 장점이 있다. 이것은 종래의 CL data로 교선을 구하는 방법과는 구별되는데 CL data의 교선을 구하는 방법은 data들을 하나의 곡선으로 취급하여 계산을 행한 것이지만 SSI 알고리즘은 offset면이 공구거동의 척도가 될 수 있다는 성질을 이용하여 offset면의 교차를 검색한 것으로 3차원 loop에 적용하기에 아무런 문제가 없다고 할 수 있다. Fig. 3의 (a)와 (b)는 각각 2차원 교선과 3차원 교선을 보여준다. Fig. 4와 5는 빠른 시간내에 교차곡선을 정의하기 위해

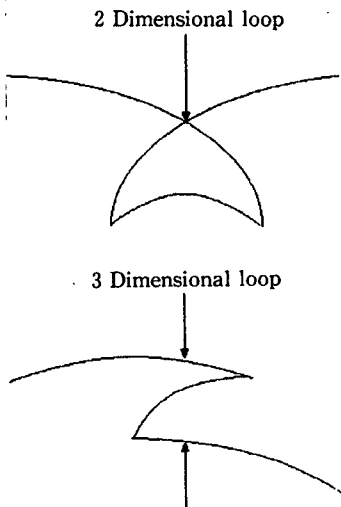


Fig. 3 2-dimensional loop and 3-dimensional loop

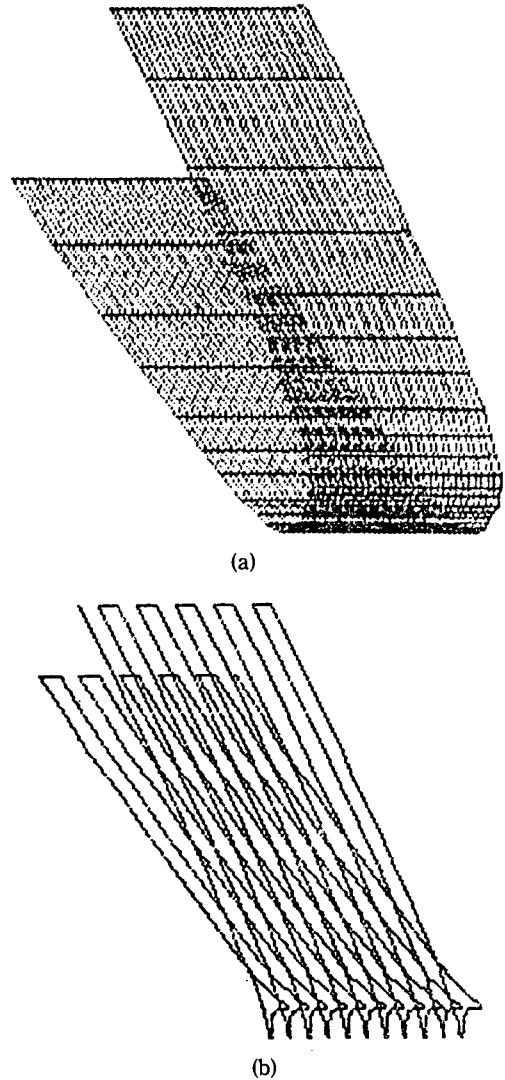


Fig. 4 Base surface and offset surface

SSI 알고리즘을 적용할 patch들을 찾아내는 원리를 보여준다. Fig. 4의 (a)와 같은 Bezier 곡면의 경우 큰 공구반경을 가진 공구로 가공할 경우 Fig. 4의 (b)와 같이 loop가 발생하는 공구경로가 생성되는데, 이 경우 Fig. 4의 (a)를 만들기 위해 사용된 입력 data들만을 공구반경만큼 offset시켜 주어서 그것을 입력 data로 하여 Bezier 곡면을 만들어 줄 경우 Fig. 5의 (a)와 같은 형태의 Bezier 조정점을 갖는 Bezier 곡면이 생성된다. 그리고 이 조정점들로 만들어지는 Bezier 곡면을 Fig. 4의

(b)에서와 같은 간격으로 분할해 보면 Fig. 4의 (b)와 Fig. 5의 (b)는 같은 모양을 나타낼 수 있다. 그리고 Bezier 곡면은 조정점들로 이루어지는 특성다각형안에서 생성되므로 이들 특성다각형이 겹치는 Bezier patch 쌍들에 한해서 SSI가 발생할 가능성이 있다. 그러므로 Fig. 4의 (b)와 Fig. 5의 (b)가 비슷한 성질을 이용해 offset면에 대해 SSI 알고리즘을 적용할 때 우선 입력 data만을 offset 시켜주어서 그것으로 만든 Fig. 5의 (a)와 같은 방식으로 생성했던 Bezier 조정점들이 이루는 특성다각형의 교차검사를 먼저 실시하여 SSI 알고리즘을 적용할 patch쌍들을 판별해 낸다. 자유곡면은 offset를 시킬 경우 invariant한 특성을 유지하지 못하므로 Fig. 4(b)와 Fig. 5(b)는 정확히 일치하지는 않다. 그러나 근사한 값을 가지므로 특성다각형을 만들어 교차 가능성을 검색하는데만 이용하여 빠른 검색을 시도하였다.

Offset 곡면을 이용하여 간섭을 검색하는 SSI 알고리즘의 적용에 있어서 고려되어야 할 사항중의 하나가 간섭이 일어나는 형태, 합성곡면인 경우와 복합곡면인 경우의 공구경로생성의 차이점이다. 본 연구에서는 기본적으로 사용되는 합성곡면인 경우에 국한을 시켜 연구를 수행하였으며 교선 loop가 발생하는 경우는 다음과 같이 분류할 수 있다.

첫째, patch와 patch간에 발생하는 loop의 경우, 측정점을 offset시켜 형성된 Bezier 특성다각형의 교차검사를 통해 검색될 수 있다.

둘째, 하나의 patch에서 이루어지는 self-intersection의 경우, 한 패치내에서 매개변수의 값의 차에 비해 그 매개변수에 해당하는 함수의 값들의 차가 아주 적을 때는 self-intersection부분이다.

이 두가지 방법을 사용해 교차가능 영역을 구하고 나면 다음의 SSI 알고리즘을 적용해 교선을 계산한다.

4. SSI 알고리즘

기하학적 곡면을 위한 알고리즘은 물체의 표현식에 의해 영향을 받는다. 교차곡선도 이러한 영향을 많이 받는다. Bezier 곡면으로 표현한 물체에 대해서는 Bezier 곡면에서 사용되는 교차곡선 알고리즘을 사용하는 제약이 받는다. 이러한 알고리즘은 그 경우에는 매우 효율적일 것이다. 그러나 다른 곡면식에 적용할 때는 응용하지 못하게 된다. SSI 알

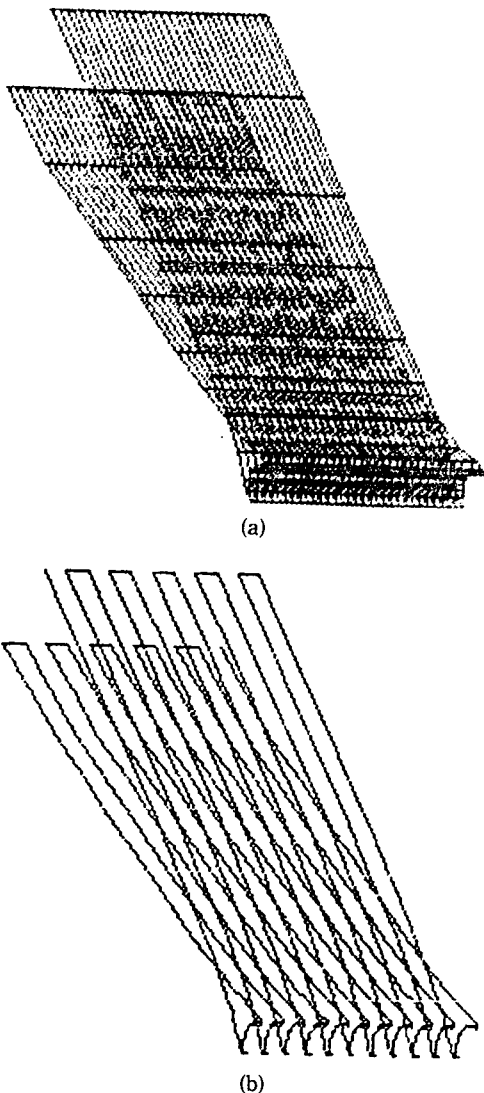


Fig. 5 Bezier surface created by control points offsetting input data and offset data

고리즘을 실제적인 system에 적용할 때에 각 곡면식에 대해 다른 알고리즘을 일일이 만드는 것은 매우 불편하다. 그래서 본 논문에서는 음함수 식과 매개변수 곡면식에서도 사용할 수 있는 일반적인 알고리즘을 이용한다. 연속이고 임의점에서 1차 미분가능한 곡면이면 모두 적용할 수 있다. 본 연구에서 개발한 SSI 알고리즘의 처리순서는 시작점 찾기, 추적 그리고 교차곡선으로 이루어져 있다.⁽⁹⁾

4.1 시작점 찾기

시작점은 교차곡선을 찾는데 있어서 최초로 사용되는 점이다. 여기에서 사용하는 방법은 격자법⁽⁷⁾을 이용하였다. 먼저 기저곡면들을 임의 개수인 격자로 나누어 서로 다른 곡면상의 격자점들의 거리를 계산한다. 격자간 거리가 곡면값과 곡면의 미분값을 이용한 Taylor 1차항을 전개한 수치보다 적으면 가능성있는 시작점으로 선택한다. Taylor 전

$$K = \max \left\{ \max_{x \in \text{conv}(x_0, x^*)} \|Df(x)\|_1, \max_{y \in \text{conv}(y_0, y^*)} \|Dg(y)\|_1 \right\}$$

$$\|A\|_1 = \max_{1 \leq j, k \leq 2} \sum_{i=1}^3 |a_{ij}| \text{ for } A \in R^{3 \times 2}$$

여기서, 이용된 norm은 Frobenius norm⁽⁷⁾이다. 그리고 $f(x_0)$, $g(y_0)$ 에서의 접평면의 교차각 α 를 고려해 더 정확한 조건식을 식(8)과 같이 얻을 수 있다.

$$\|f(x_0) - g(y_0)\|_1 \leq c \cdot H \cdot M \sin \alpha \quad (8)$$

여기서, c 는 상수이고 M 은 다음과 같다.

$$M = \max(\|D\phi(x)\|), \phi \in \{f, g\}, x \in G_0$$

점차적으로 H 을 줄여가며 0(zero)으로 가도록 반복하면 정확한 교차점에 아주 근사한 점들을 찾을 수 있다.

이 알고리즘은 self-intersection를 얻는데도 유용하게 이용될 수 있다. 이 경우에는 함수 f, g 를 동일한 함수로 놓는다. $\|f(x_0) - g(y_0)\|$ 은 매우 작지만 두 매개변수 값의 $\|x_0 - y_0\|$ 이 상당히 큰 점이 self-intersection점으로 보장된다.

4.2 추적

추적(tracing)은 시작점을 시작으로 연속적으로 교차곡선상의 점들을 찾아가는 것이다. 시작점에서의 곡면들의 법선을 이용하여 추적의 방향을 정하고 적당한 step벡터(추적의 거리)의 거리를 준다.

개식의 1차항까지 이용한 근사값이므로 더 정확한 값을 구하기 위해서 격자점들의 법선벡터를 이용한 교차각을 이용하여 정확한 시작점 쌍을 구한다. 그리고 선택된 점들을 중심으로 처음의 격자거리보다 더 미세하게 다시 나누어서 정확한 시작점을 찾는다. 나누는 격자간격이 0에 가까우면 정확한 시작점을 구하게 된다. 시작점을 구하는 수식은 다음과 같다.

$n \in N, H=1/n, G_0$ 는 정의구역의 값들이라 하자. N 은 정수이며 n 은 곡면을 나누는 수이고 $(n+1) \times (n+1)$ 은 격자들의 수이다. 교차점들 x^*, y^* 에 대하여 $f(x^*) = g(y^*)$ 이 된다. 밑의 조건을 만족시키는 한 쌍의 점 x_0, y_0 가 G_0 의 domain에 존재하게 되면 다음 식을 만족한다.

$$\|f(x_0) - g(y_0)\|_1 \leq 2HK \quad (7)$$

여기서,

만약 step의 길이가 너무 크다면 refinement 시간이 많이 걸리고 오차가 크게 나타난다. step의 길이가 너무 작다면 추적속도가 느려지게 되어 알고리즘의 효율성이 감소한다. 여기에서는 교차점에서의 곡률을 근거로 적용성이 있는 step 길이를 이용한다. 곡률이 심하면 step의 길이가 짧아지고 곡률이 완만하면 step의 길이가 커져서 효율성을 높게 된다. 여기에서 곡률계산은 효율성을 위해서 2차 미분을 구하지 않고 Faux와 Pratt⁽¹⁾가 제시한 근사곡률을 이용하였고 식은 아래와 같다.

3점 P, Q, R 가 주어질 때에 근사곡률반경 ρ 은 다음과 같다.

$$\rho = \frac{|a||b||a-b|}{2|a \times b|} \quad (9)$$

여기서, $a = Q - P, b = R - P$ 이다.

Step 길이를 δ 라 하고, δ 는 $\Delta\theta \cdot \rho$ 로 표현된다. $\Delta\theta$ 은 미소각이고 여기서는 0.1 rad로 정하였다. 1차 미분이 0(zero)이거나 두 곡면의 접평면이 동일하면 분기점이다. 이 점에서는 특별한 추적이 필요하다.

Step 길이 δ 을 구한 후에는 추적을 한다. 추적에는 근사추적점을 구하는 것과 정확한 추적점을

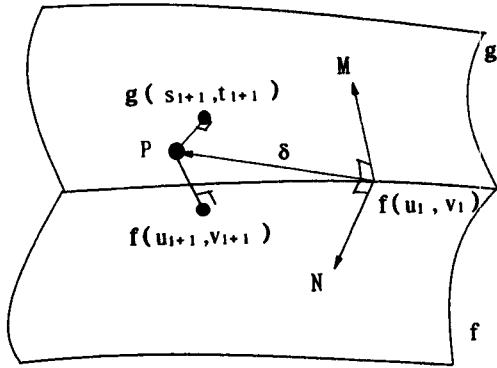


Fig. 6 Determination of initial guess point

구하는 point refinement로 나누어 생각할 수 있다. 먼저 근사추적점에 관해서 설명하고자 한다. Fig. 6에서 보이는 와와 같이 첫 추적점 P는 교차곡선을 따라 만들어져야 한다. 만약 추적점 P가 step 길이가 δ 이고 현재 시작점에서 교차곡선의 접선방향으로 위치해 있다면 P의 위치는 (10)식과 같다.

$$P = f \pm \delta(M \times N) / |M \times N| \quad (10)$$

위의 식에서 부호는 추적방향을 의미한다. 여기서 M, N는 곡면 f, g의 단위법선벡터이다. 다음 단계는 추적점 P에 대한 정의의 구역값을 Jacobian inverting하여 구하면 식(11)과 같다.

$$\left[\frac{\partial f}{\partial u} \Big|_{u_i, v_i} \quad \frac{\partial f}{\partial v} \Big|_{u_i, v_i} \right] \begin{bmatrix} u_{i+1} - u_i \\ v_{i+1} - v_i \end{bmatrix} = [P - f(u_i, v_i)] \quad (11)$$

여기서, 벡터는 행벡터이다. 왼쪽 항의 3×2 행렬은 Jacobian 변환이다. 즉, 이 식은 근사추적점 P와 곡면 f사이의 최소거리에 해당하는 곡면 f상의 매개변수 (u_{i+1}, v_{i+1}) 를 구한다. 곡면 g와 P사이의 최단거리의 점은 식(11)에서 f를 g로 대체하면 된다. 두 점이 SPT(same point tolerance)내에 속하지 않는다면 point refinement를 반복 수행한다.

이 추적점은 근사점이므로 정확한 교차곡선상의 점이 아닐 수 있다. 근사한 교차곡선의 점들을 정확한 교차곡선상의 점들로 보내는 과정을 'point refinement'라고 한다. 추적에서 구한 근사점을 Jacobian inverse를 이용하여 추적점과 곡면상에 최단거리에 있는 점을 구하고 그 점에서 접평면을 만든다. 그리고 그 접평면들의 법선벡터를 이용하

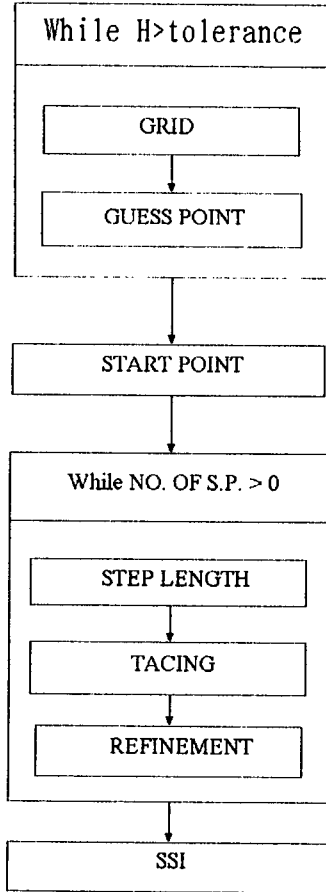


Fig. 7 Process of SSI algorithm

여 또 하나의 평면을 만들어 3평면을 교차하면 교차곡선에 보다 가까운 점을 얻을 수 있다. 원하는 공차까지 이 방법을 계속하면 된다.

Point refinement는 위식에서 구한 근사점들을 교차곡선상에 보내어 실제 곡선에 더 가까이 놓는 것이다. 근사점에서의 평면 f, g의 법선을 M, N으로, 또한 MN을 $M \times N$ 으로 표현하고, M, N이 이루는 평면의 법선으로 하면 교차점은 다음과 같이 계산된다.

$$\begin{bmatrix} M^T \\ N^T \\ MN^T \end{bmatrix} [P_{i+1}] = \begin{bmatrix} M \cdot f \\ N \cdot g \\ MN \cdot (f+g)/2 \end{bmatrix} \quad (12)$$

추적은 다음의 종료조건 중 하나라도 만나면 끝난다.

① 추적이 처음 시작점과 만날 때, ② 추적에서 구해진 매개변수 값 쌍이 한계치를 벗어날 때, ③

추적에서 다음 값이 정해진 증가치 이내일 때, 만약 약에 종료조건 ①의 경우라면 교차곡선은 패곡선이 된다. 그렇지 않으면 개곡선 교차곡선을 얻게 된다. ②에서 매개변수의 범위는 $[0, 1]$ 이다. ③의 조건은 하나의 patch 안에서 정의되는 self-intersection의 경우에 적용되며 여기서는 매개변수의 값들의 차는 증가하는데 곡면의 값들의 차는 감소하는 경우이다. 증가치는 매개변수와와 차와 증가치의 비로 결정되는 것으로 매개변수의 차가 0.1 이상인데 곡면값의 차가 0.001(본 논문에서는 같은 점이라고 인정하는 공차)이하일 때 self-intersection으로 간주하였다.

Fig. 7은 위에서 설명한 과정을 아래의 순서도로 간략하게 나타냈다.

5. SSI 알고리즘의 적용

SSI 알고리즘을 적용해 교선군을 구하기 위해 우선 교차가 일어날 가능성이 있는 patch 쌍들을 빠르게 구해주어 계산시간을 단축할 필요가 있다. 이를 위해 측정점들을 법선방향으로 공구반경만큼 offset시켜 얻은 data들로 Bezier 조정점들을 생성하고 그 조정점들로 Bezier 곡면을 생성하여, 이를 원래의 입력 data에서 얻은 offset 곡면과 비교해보면 거의 근사한 것을 알 수 있다. 이를 이용해 교차가능 patch들의 쌍을 신속하게 근사적으로 구할 수 있음을 알 수 있다. 즉 Bezier 곡면은 그 조정점들로 이루어지는 특성다각형 안쪽에서 특성다각형의 네 모서리를 지나며 형성되므로 교차가 일어날 가능성이 있는 부분에서는 각 특성다각형끼리의 교차가 발생할 것이다. 이때 원래의 입력 data들로 만들어진 offset 곡면은, 원래의 입력 data들을 공구반경만큼 offset시킨 data들로 만들어진 offset data들과 근사적으로 일치하므로 빠른 시간 안에 계산이 가능한 이 원래의 입력 data들을 공구반경만큼 offset시켜 구한 Bezier 조정점들로 이루어지는 특성다각형의 교차판별과 self-intersection 검사를 통해 교차가능 영역의 판정을 신속하게 행한다.

교차가능성이 있는 곡면 patch쌍들과 self-intersection이 발생할 가능성이 있는 patch가 구해지면 이 각각의 patch를 하나의 곡면으로 가정해 위의 교선산출 알고리즘을 적용한다. 이때 SSI 알고리즘의 수행을 위해서는 교차하는 두 곡면의 매개변

수 표현식과 그 식의 u, v 방향 1차 미분식이 필요하게 된다. 본 논문에서 사용하는 알고리즘에는 곡면의 기저식으로 Bezier 곡면을 사용하였으므로, offset 곡면을 형성하는데 있어서 필요한 data들은 교차가능한 patch쌍들을 찾는 계산에서 미리 구해져 있다. 그러므로 공구 반경만 지정되면 그에 해당하는 교선 data가 계산되어진다. Bezier 곡면은 본 실험실에서 개발한 surface modeler에 의해서 생성하였다. 교선 data가 구해지면 다음으로 교선을 공유하는 두 평면에서 공구경로를 정의해 주어야 한다.

본 논문에서는 3축 제어 NC milling을 위한 공구경로를 고려하였고 parametric 경로를 사용하므로 공작물에 대한 공구의 접근은 공구중심을 지나 Z축의 방향으로 parametric 경로를 따른다. 그러므로 교차가 발생하는 patch간에 있어서 먼저 만나는 patch에서는 그 patch의 경로를 따르다가 교선과 만나는 경우 다음 patch의 가공경로를 따라 가공을 수행해 나간다. Fig. 8의 (a)는 기저곡면과 간섭 loop가 발생하는 offset 곡면을 보여주

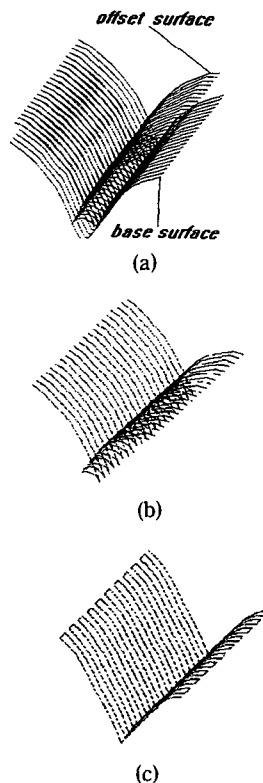


Fig. 8 Simple example for interference avoidance

고 Fig. 8의 (b)와 (c)는 offset 곡면과 간섭 loop가 제거된 예를 보이고 있다.

6. Program 수행과 절삭 예

6.1 Program의 구성

공구간섭제거 알고리즘은 다음 순서도에 따라 구성되어 있다. Program은 Pascal를 사용하여 구현되었고 각 단계의 수행은 다음과 같다.

(1) 모델링 data는 3차원 측정기를 통하여 얻어졌다.

(2) 기저곡면을 Bezier 곡면으로 생성하였다.

(3) data점들을 offset시킨 Bezier 조정점들을 형성한 후 특성다각형을 만들어 교차가능성 및 self-intersection 가능성 검색을 하였다.

(4) 교차가능성이 있는 patch군에 대해 SSI 알고리즘을 적용한 정확한 교차곡선을 찾아내었다.

(5) offset 곡면의 loop를 제거하고 공구경로(CL)를 계산하였다.

(6) 가공경로를 따라 황삭가공을 수행하였다.

6.2 Program의 수행 예

6.2.1 Data의 측정 및 구조

Data의 측정은 3차원 측정기를 사용하여 측정하였다. 가공 대상물은 CD 카세트 플레이어의 리모콘을 대상으로 하였는데 이는 실제 사용되는 물건으로 보통 많이 보이는 리모콘같이 딱딱한 사각입방체형이 아닌 부드러운 심미적 곡면을 가지고 있을 뿐만 아니라 적당한 공구로 가공하면 공구간섭이 발생하는 형태를 취하고 있기 때문이다. 아직은 본 연구에서 수행된 프로그램이 복합곡면이 아닌 합성곡면으로 전체의 형상을 하나의 곡면으로 모델링 했다. 기존의 연구(10)를 보면 data 격자의 배열이 불규칙적이라도 모델링의 오차발생이 오백분의 일정도이므로 데이터의 측정시 측정간격의 불균일성에 대한 사항은 별로 고려하지 않았으나, 균일한 간격을 유지해 준 경우 오차가 더욱 작기 때문에 가끔씩 균일한 간격으로 측정을 하여 모델링 오차를 줄였다. Fig. 10은 입력 data의 구조를 보여준다.

6.2.2 Program의 수행결과

측정 data들을 입력하여 모델링을 수행시켜 다

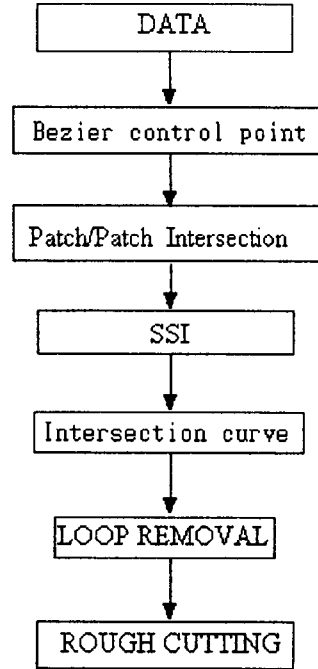


Fig. 9 Process of interference avoidance algorithm

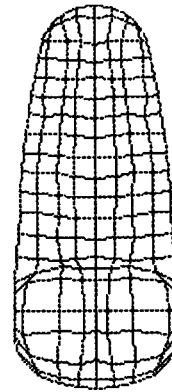


Fig. 10 Structure of input data

음과 같은 결과들을 얻었다. Fig. 11은 반경 7.5 mm의 공구로 가공을 수행하였을 경우의 offset 경로를 보여준다. 이 경우에서 보는 바와 같이 곡률이 급격히 변하는 오목부위에서 간섭 loop가 발생함을 알 수 있다. Fig. 12에서 (a1)과 (b1)은 SSI 알고리즘을 사용한 Fig. 11의 3, 4번째 공구경로의 loop가 발생한 모습이고 (a2)와 (b2)는 각각 loop가 제거된 모습을 보이고 있다. Fig. 13은 간섭 loop를 제거한 offset 경로를 보여준다.

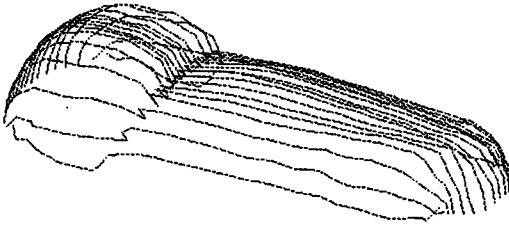


Fig. 11 Offset path with interference

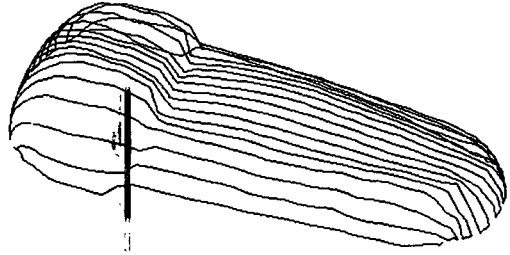


Fig. 13 Interference-free offset path

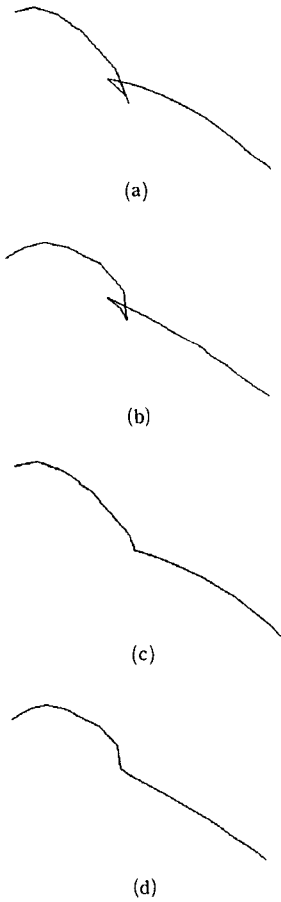


Fig. 12 Process of interference check

Fig. 14의 (a)와 (b)는 각각 Fig. 11과 13의 간섭 발생부위와 간섭이 제거된 부위를 확대한 그림이다. Fig. 14의 (a)에 보이는 3차원 loop는 Fig. 3에 보이는 것과 비슷함을 알 수 있다.

Parametric 가공경로에서는 양방향(u, v)의 경로중 한 방향이라도 공구간섭이 발생하면 공구간섭 부위이므로 양방향으로 검사를 행했고 여기에서는

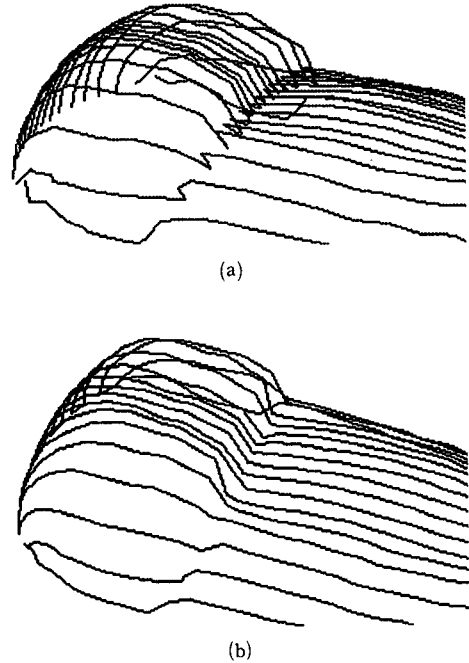


Fig. 14 Magnified zone of interference occurrence and interference-free region in Fig. 11 and 13 respectively

u 방향 공구경로에서 공구간섭이 발생했다.

6.3 가공재료, 가공공구, 가공기계

본 연구에서는 아크릴재료를 사용하여 CD용 리 모콘의 하단부 모형을 공구반경 7.5 mm의 ball mill을 이용해 이송 150 mm/min, 주축회전속도 1400 rpm으로 간섭을 제거한 공구경로로 황삭가공을 수행하였다. 대개는 cusp의 크기 증가와 사상 작업을 위해 곡률이 큰 방향으로 가공을 행하는데 여기서는 가공속도를 중시하는 황삭이라 길이방향

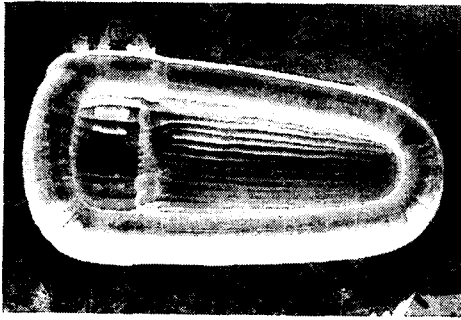


Fig. 15 A simple example of machined part

으로 가공경로를 잡았고 길이방향 경로에서 공구간섭이 검색되었기 때문에 길이방향 경로를 이용하여 가공하였다. 본 연구의 목적이 간섭이 발생하는 공구에 의한 간섭을 제거한 공구경로생성이므로, 다른 절삭조건은 일정하게 생각했다.

SSI 알고리즘을 적용하여 공구간섭을 제거한 공구경로를 생성하고 post processing하여 NC code를 생성하여 DNC 시스템에 의하여 가공한 결과를 Fig. 15에서 보인 바와 같이 공구간섭방지 software의 실용성을 입증하였다. 또한 가공결과 가공시에 남게 되는 cusp는 직경이 큰 공구를 사용할 경우 감소되는데 정삭의 경우에도 간섭이 발생하지 않는 공구를 위해 너무 작은 반경의 공구를 택하지 않고도 보다 큰 공구로 간섭이 없는 공구경로를 생성해 필요한 부분의 추가절삭을 행하는 것도 좋으리라고 사료된다.

7. 결 론

공구간섭을 제거한 공구경로의 생성을 위하여 SSI 알고리즘을 적용한 교차검색방법에 의해 구현할 수 있는 간섭검색 및 방지 software를 개발하였으며 이 software를 통하여 간단한 대상물을 택해 간섭 없는 황삭가공경로를 생성하여 가공을 수행한 결과 다음과 같은 결론을 얻을 수 있었다.

(1) 공구간섭 가능한 patch쌍과 self-intersection 발생가능한 영역의 검사에는 주어진 data들의 offset점들을 이용한 Bezier 곡면의 특성다각형간의 교차가능성 판별을 이용하여 구했다.

(2) CL data 상에서 3차원적으로 loop가 발생하는 경우에도 각각의 patch에서의 parameter의 값으로서 그 교점의 위치를 정확히 계산해 낼 수 있으므로 경로생성의 신뢰성을 높일 수 있었다.

(3) 교차곡선은 주어진 입력 data들의 offset점들만으로 구해질 수 있으므로 parametric 공구경로의 경우 곡면 전체의 CL data를 미리 구하지 않고도 가공하고자 하는 방향으로의 patch열별로 CL data를 구해서 가공을 수행할 수 있게 되어 필요한 기억용량을 크게 줄일 수 있게 되었다.

후 기

본 연구는 1993년도 교육부 학술연구조성비(기계공학 ME93-E-17)의 지원을 받아 수행되었으며 이에 관계자에게 감사드립니다.

참 고 문 헌

- (1) Faux, I. D. and Pratt, M. J., 1981, *Computational Geometry for Design & Manufacture*, Ellis Horwood Pub.
- (2) Choi, B. K., Lee, C. S., Hwang, J. S. and Jun, C. S., 1988, "Compound Surface Modelling and Machining," *CAD*, Vol. 20, No. 3, pp. 127~135.
- (3) Duncan, J. P. and Mair, S. G., 1977, "The Anti-Interference Features of Polyhedral Machining," *Advances in Computer-Aided Manufacture*(McPherson, D. ed.), North-Holland Pub, pp. 181~195.
- (4) Duncan, J. P. and Mair, S. G., 1983, "Sculptured Surfaces in Engineering and Medicine," Cambridge Univ. Press.
- (5) Choi, B. K. and Jun, C. S., 1989, "Ball-End Cutter Interference Avoidance in NC Machining of Sculptured Surface," *CAD*, Vol. 21, No. 6, pp. 371~378.
- (6) Kondo, K., Kishinami, T. and Saito, K., 1988, "Machining System Based on Inverse Offset Method," *Journal of JSPE*, Vol. 54, pp. 971~976.
- (7) Gregor Mullenheim, 1991, "On Determining Start Points for a Surface/Surface Intersection Algorithm," *CAGD*, Vol. 8, pp. 401~408.
- (8) Barnhill, R. E. and Kersey, S. N., 1990, "A Marching Method for Parametric Surface/Surface Intersection," *CAGD*, Vol. 7, pp. 257~280.
- (9) 양균의, 이희관, 1993, "교차곡선을 이용한 GC' 블렌딩형성에 관한 연구," 대한기계학회 '93

년도 추계학술대회 논문집, pp. 653~657.

면 모델링에 관한 연구,” 전북대학교, 공학연구

(10) 양균의, 성원정, 1991, “NC가공을 위한 곡

제22집, pp. 281~290.